



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Am

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/838,478	04/19/2001	Jens Krause	CH920000020US1	8899

7590 05/31/2005

Gregory M. Doudnikoff
IBM Corporation T81//503
Research
PO Box 12195
Triangle Park, NC 27709

EXAMINER

PHAM, CHRYSTINE

ART UNIT PAPER NUMBER

2192

DATE MAILED: 05/31/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/838,478

Applicant(s)

KRAUSE, JENS

Examiner

Chrystine Pham

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 29 December 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-12 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-12 is/are rejected.
- 7) ☒ Claim(s) 11 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

DETAILED ACTION

1. This action is responsive to Amendment filed on December 29th 2004. Claims 1, 2, 7, 8, 11, and 12 have been amended. Claims 1-12 are presented for examination.

Response to Amendment

2. In view of the amendment to the specification to correct the identified grammatical error, the objection to the specification is hereby withdrawn.
3. In view of the amendment to claims 2, and 12 to correct the identified informalities, objection to claims 2, and 12 is hereby withdrawn.
4. In view of the amendment to claims 2, 7, 8, 11 to overcome rejection under 35 U.S.C. 112, second paragraph, rejection under 35 U.S.C. 112 of claims 2-3, 7, 8, 11 is hereby withdrawn.

Response to Arguments

5. Applicants' arguments filed December 29th 2004 have been fully considered but they are not persuasive.

The Applicants essentially contend that "Cirne discloses a system for modifying object oriented code" which is "fundamentally different from the present invention as recited in the present claims" (page 9, section B).

First, the Applicants contend that the cited portions of Cirne (col.4:34-42; col.5:4-7) do not teach "converting at least one said class field to an instance field and introducing the instance field into said helper class" (page 9, section B). It is submitted that, in col.4:34-42, Cirne teaches calling the code modifier 10 to substitute/replace an original static field (i.e., class field) with a new static field (i.e., instance field). The new static field is defined in a subclass (i.e., helper class) of the class defining the

original static field (i.e., original class). Since the code modifier 10 is responsible for changing/modifying the code associated with the left parameter (i.e., original static field) of a substitute rule, substituting/replacing the original static field with a new static field, which is defined in a subclass of the class defining the original static field, is the equivalence of "converting at least one said class field to an instance field and introducing the instance field into said helper class". With respect to col.5:4-7, the passage indeed discusses changing a static field by creating a static field change rule as the Applicants have pointed out. Changing a static field, by creating a static field change rule, which will be input to code modifier 10 so that a new static field can be defined only reinforces what has been taught in col.4:34-42.

Second, the Applicants assert that col.3:65-4:3 of Cirne does not teach "converting the original-class class-initialization method to a helper-class instance-initialization method" (page 10). The Applicants also state that "this passage describes a substitute class rule to change the code that allocates an object of a first class to code that allocates the object of a new class" (page 10). It is submitted this very passage clearly teaches "converting the original-class class-initialization method to a helper-class instance-initialization method" since the code that allocates an object of a first class is the equivalence of the "original-class class-initialization method" and code that allocates the object of a new class is the equivalence of the "helper-class instance-initialization method". Thus, changing the code that allocates an object of a first class to code that allocates the object of a new class is the equivalence of changing or "converting" an "original-class class-initialization method" to a "helper-class instance-initialization method". In col.4:2, Cirne specifically discloses the "new class" as a subclass of the original class, with the former being the equivalence of the "helper-class" as established above. The Applicants are referred to col.4:10-12, in which Cirne specifically discloses a constructor method (i.e., method <init>) for allocating or initializing an object of a class. Thus, code for allocating an object of a class IS indeed the equivalence of the claimed "initialization method". And again, changing the code that allocates an object of a first class to code that allocates the object of a new class is the equivalence of changing or "converting" an "original-class class-initialization method" to a "helper-class instance-initialization method".

Art Unit: 2192

Third, the Applicants contend that col.4:10-14 does not teach introducing it (i.e., helper-class instance-initialization method) into said helper class which comprises a helper-class class-initialization method. It is submitted that, in Java, every class comprises a "class-initialization method" (i.e., constructor method <init>) so that objects of the class can be initialized/instantiated. Furthermore, as established above, the original class has a constructor method or "class-initialization method" (i.e., code that allocates an object of a first class), thus, the subclass (i.e., helper class) inherently comprises a "class-initialization method" (i.e., constructor method <init>), which it inherits from the original class. Contrary to Applicants' argument, this very passage clearly teaches "introducing it into said helper class which comprises a helper-class class-initialization method".

6. In view of the foregoing discussion, rejection of claims 1-12 under 35 U.S.C. 102(b) is considered proper and maintained.

Claim Objections

7. Claim 11 is objected to because of the following informalities: "a usage message" (line 3) should be changed to --a usage method-- in order to provide an antecedent basis for "the usage method" recited in line 12. Appropriate correction is required.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1-12 are rejected under 35 U.S.C. 102(b) as being anticipated by Cirne (U.S. Patent 6,260,187) (hereinafter *Cirne*).

As per claim 1, *Cirne* teaches a method of transforming a class (e.g., FIG.1 & associated text), comprising a usage method accessing at least one class field (e.g., see <original class>, <original static field> col.3:65-col.4:45; see class *Button*, new class *RedButton*, *setBackground* col.14:55-col.15:60) said class being loadable by a class loader in an object-oriented environment, said method comprising the steps of:

(a) creating from an original class, which comprises a class field (e.g., see <original class>, <original static field> col.3:65-col.4:45), an original-class class-initialization method (e.g., see <original class>, code that allocates an object of the <original class>, constructor, method <init> col.3:65-col.4:45), and a helper class (e.g., see FIG. 2 & associated text; see <original class>, code that allocates an object of the <original class>, <new class>, subclass, col.3:65-col.4:45), by

i) converting at least one said class field to an instance field and introducing the instance field into said helper class; (e.g., see <original static field>, <new static field>, subclass col.4:34-42, and col.5:4-7) and
ii) converting the original-class class-initialization method to a helper-class instance-initialization method (e.g., code that allocates an object of the <original class>, code that allocates the object to be of the <new class> col.3:65-col.4:3) and introducing it into said helper class which comprises a helper-class class-initialization method (e.g., see <new class>, subclass, methods, fields, constructor, method <init> col.4:1-14);
and

(b) creating for the class a corresponding modified class (i.e., modified-original class) by converting the usage method of the original class to a modified-usage method and introducing it into said modified class, wherein each access to the

class field is replaced by an invocation of an access function for fetching, for a process with an instance of the helper class, from the instance, the instance field corresponding to the class field (e.g., col.5:4-11, col.17:33, FIG.6 step 350 & associated text), the helper class and the modified class being loadable by the class loader (see Java Virtual Machine col.6:49; see 80 FIG.4 & associated text; FIGS.7,8 & associated text).

As per claim 2, it recites limitations which have been addressed in claim 1, therefore, is rejected for the same reasons as cited in claim 1.

As per claim 3, *Cirne* teaches a method as applied to claim 2, wherein said creating step (c) comprises creating, for each class field in the original class, at least one of an access function, a read access function (e.g., col.17:33) and a write access function (e.g., col.15:46).

As per claims 4-5, they recite limitations which have been addressed in claim 1, therefore, are rejected for the same reasons as cited in claim 1.

As per claim 6, *Cirne* teaches a method as applied to claim 1, wherein transforming the class is applied to a byte code (e.g., col.21:1-2).

As per claim 7, *Cirne* teaches a method as applied to claim 1, further comprising the step of loading the helper class and the modified class by use of the class loader when the process is started (see Java Virtual Machine col.6:49-50 & pg.3 line 1-9 of the applicant's specification).

As per claim 8, *Cirne* teaches a method as applied to claim 2, wherein said converting substep (ii) further comprises introducing the original-class class-initialization method into the modified-original class and replacing the original-class class-initialization by an empty method (e.g., col.12:36-41).

As per claim 9, *Cirne* teaches a method as applied to claim 1, wherein the helper-class class-initialization method creates a table (e.g., col.11:18-20, col.6:5-9, col.14:11-15).

As per claim 10, *Cirne* teaches a method as applied to claim 1, further comprising the step of transforming an original interface, comprising at least one class field and/or an original-interface class-initialization method (e.g., col.6:37-40 and col.6:55-56) into a modified interface and the helper class. Claim 10 also recites limitations which have been addressed in claim 1, therefore is rejected for the same reasons as cited in claim 1.

As per claim 11, *Cirne* teaches a computer readable code stored on computer readable media (e.g., col.3:1-9, col.17:56-60, col.18:12-14) for transforming a class in an object-oriented environment (e.g., see <original class>, <original static field> col.3:65-col.4:45; see class *Button*, new class *RedButton*, *setBackground* col.14:55-col.15:60), comprising:

a first process for creating from an original class, which comprises a class field and a usage method for accessing the class field (e.g., see *fields* 224 in FIG.5 & associated text; see <original class>, <original static field> col.3:65-col.4:45; see class *Button*, new class *RedButton*, *setBackground* col.14:55-col.15:60), an original-class class-initialization method (e.g., see *methods* 228 in FIG.5 & associated text; see <original class>, code that allocates an object of the <original class>, constructor, method <init> col.3:65-col.4:45), and a helper class (e.g., see FIG. 2 & associated text; see <original class>, code that allocates an

object of the <original class>, <new class>, subclass, col.3:65-col.4:45), said first process (e.g., FIG. 1) comprising:

first subprocesses for converting at least one said class field to an instance field and introducing the instance field into the helper class (e.g., see <original static field>, <new static field>, subclass col.4:34-42, and col.5:4-7);

and second subprocesses for converting the original-class class-initialization method to a helper-class instance-initialization method (e.g., code that allocates an object of the <original class>, code that allocates the object to be of the <new class> col.3:65-col.4:3) and introducing it into the helper class which comprises a helper-class class-initialization method (e.g., see <new class>, subclass, methods, fields, constructor, method <init> col.4:1-14); and

a second process for creating for the class a corresponding modified class by converting the usage method to a modified-usage method, wherein each access to the class field is replaced by an invocation of an access function for fetching, for a process with an instance of the helper class, from the instance, the instance field corresponding to the class field (e.g., col.5:4-11, col.17:33, FIG.6 step 350 & associated text), the helper class and the modified class being loadable the class loader (see Java Virtual Machine col.6:49-50; see 80 FIG.4 & associated text; FIGS.7,8 & associated text).

As per claim 12, *Cirne* teaches, in a computing environment, a system for class transformation, said system comprising:

a class comprising at least one class field (e.g., see *<original class>*, *<original static field>* col.3:65-col.4:45; see *class Button*, *new class RedButton*, *setBackground* col.14:55-col.15:60), an original-class class-initialization method (e.g., see *methods 228* in FIG.5 & associated text; see *<original class>*, *code that allocates an object of the <original class>*, *constructor*, *method <init>* col.3:65-col.4:45), and a usage method accessing at least one of the class fields (e.g., see *class Button*, *new class RedButton*, *setBackground* col.14:55-col.15:60), said class residing in memory (e.g., col.3:56-60); and

a creator module for creating, out of said class, a helper class and a modified class (e.g., see FIG. 2 & associated text; see *code modifier 10*, *<original class>*, *code that allocates an object of the <original class>*, *<new class>*, *subclass*, col.3:65-col.4:45),

wherein said at least one class field is convertible to an instance field into said helper class (e.g., see *<original static field>*, *<new static field>*, *subclass* col.4:34-42, and col.5:4-7), wherein said original-class class-initialization method is convertible to a helper-class instance-initialization method (e.g., *code that allocates an object of the <original class>*, *code that allocates the object to be of the <new class>* col.3:65-col.4:3) into said helper class which comprises a helper-class class-initialization method (e.g., see *<new class>*, *subclass*, *methods*, *fields*, *constructor*, *method <init>* col.4:1-14), and wherein in said usage method in said modified class each access to said class field is replaceable by an invocation of an access function for fetching the instance field corresponding to the class field (e.g., col.5:4-11, col.17:33, FIG.6 step 350 & associated text) for a process with an instance of said helper class, from said

instance, and wherein said helper class and said modified class are loadable by a class loader (see Java Virtual Machine col.6:49-50; see 80 FIG.4 & associated text; FIGS.7,8 & associated text).

Conclusion

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chrystine Pham whose telephone number is 571-272-3702. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.

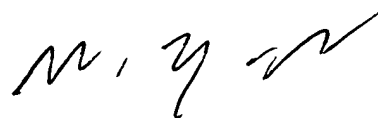
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Application/Control Number: 09/838,478
Art Unit: 2192

Page 11

CP
May 18, 2005

A handwritten signature in black ink, appearing to read 'M. Y. Zhen' with a stylized flourish at the end.

WEI Y. ZHEN
PRIMARY EXAMINER